Adaptable Authentication Model - for Exploring the Weaker Notions of Security

Ahmed, Naveed; Jensen, Christian D.

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

Adaptable Authentication Model

for Exploring the Weaker Notions of Security[1]

Naveed Ahmed **Christian D. Jensen**

Technical University of Denmark (DTU)
Institute of Informatics and Mathematical Modelling (IMM)
IMM-Technical Report-2010-17
Last Updated: 29th November 2010

Please send your comments to naah@imm.dtu.dk.

[1]An abridged version of this report appears in the Proceedings of ESSOS-2011; the original publication is available at www.springerlink.com.

Abstract

There are at least a few hundred published protocols that fall in the category of authentication and key establishment. Under a naive definition of authentication and key establishment, the existence of so many protocols is quite fascinating and somewhat stunning for a newcomer to the field of communication security. One potent argument often presented is we keep designing new protocols due the demand of new type of applications and due to the discovery of flaws in existing protocols. While designing new protocols for new type of applications, such as RFID, is definitely an important driving factor nevertheless the most among the published protocols are in fact the result of discovery of flaws in their predecessors.

As our understanding of cryptography and protocol analysis is getting mature, the ability to discover new flaws in the protocols also increases. We now have a better understanding of actual operational environment. In past, this often caused increasing the power of attacker model, for instance, now a days we also consider privacy concerns and side channel leakage beside the classic Dolev-Yao attacker.

A protocol is labeled as insecure protocol once an effective attack or flaw is found in it. In fact, the most of the published protocols are considered insecure from this point of view. In practice, however, this approach has a side effect, namely, we rarely bother to explore how much insecure is the protocol. This question asks us to explore the area between security and insecurity; after all neither a flawed protocol is always completely insecure neither all applications require the security against an all powerful attacker.

The current approach towards security analysis, which we call *strict security*, considers a protocol along with a powerful attacker, such as Dolev-Yao attacker and sometimes with additional capabilities such as dynamic corruption of communicating nodes. Then, one tries to show that the protocol achieves its objective under this specific attacker. Naturally there are three possibilities: one may succeed in constructing a security proof; one may fail in proving security, which often makes the protocol suspicious; or one may discover a concrete attack, which definitely makes the protocol insecure under such strict definition of the attacker.

There is however an alternate — *adaptable security*, which we propose as a more general approach to the security problem. The approach considers correct protocols, i.e., protocols that achieve their objectives when there exist no effective attacker. All correct protocols are assumed to be secure and the challenge we pose for a security analyst is to derive the least strongest attacker (LSA) model for which the, so-called, a priori assumption about security holds. In this way, the security definition of a protocol can be adapted to suitable choice of LSA.

Another aspect of the proposed approach is the flexible treatment of security goals; we decompose high level security goals in many fine level goals and a protocol may achieve only a subset of all fine level goals. We believe that these flexible choices of attackers and security goals are more

practical in many real world scenarios. An applications may require the protection against a weaker attacker and may require to achieve fewer security goals.

Contents

Chapter 1

Introduction

Entity authentication protocols are used to verify the 'identity' of a far-end entity. This may appear to be a simple goal, but in reality it entails establishing many fine level authentication goals (FLAGs) [4], which are listed in the following:

Existence If an entity A verifies that the peer entity once existed on the network, then A is said to achieve the goal *Existence*. Existence does not require any timeliness property or even the awareness that peer entities have achieved the goal. Existence only guarantee that an entity once existed on the network and have sent some message.

Operativeness If an entity A verifies that a peer entity currently exist on the network, then A is said to achieve the goal *operativeness*.

Willingness If an entity A verifies that once a peer entity B wanted to communicate with A, then A is said to achieve the goal *Willingness* for B. Note that there is no timeliness requirement associated with this goal.

Single-sided Authentication Single-sided authentication is achieved if an entity A verifies that the peer entity B is currently ready to communicate with A.

Confirmation If an entity A verifies that the peer entity B knows that X has achieved a security goal G (with Y as peer entity), then A is said to achieve the goal *confirmation* for G from B.

Strong Single-sided Authentication Strong single-sided authentication is achieved if an entity A has the *confirmation* for the corresponding *single-sided authentication* from B. Intuitively, it means that A not only (currently) authenticates B, but also has the assurance that B knows about this event.

Mutual Authentication If an entity A verifies that both parties (A and peer entity B) currently want to communicate with each other, then A is said to achieve *Mutual Authentication*.

Strong Mutual Authentication Strong mutual authentication is achieved if an entity A has the *confirmation* for the corresponding *mutual authentication* from B.

Sometimes it is assumed that an adversary can corrupt entities. In that case *forward* and *backward* security [9] are also important.

Traditionally, the objective of an authentication protocol designer is to provide the strongest possible security. So-called "standard authentication models" normally incorporate a quite powerful adversary and then the security analysis is aimed at verifying whether a given protocol is secure or not. The most common example is the use of Dolev-Yao attacker[1][13] in the security models used in most of the tools based on formal analysis [2, 1, 7]. Similarly, various accepted definitions of security, e.g, *correspondence* [27] or *intensional* properties [22], may implicitly include all of the aforementioned FLAGs.

Often vulnerabilities are identified in the authentication protocols, based on a security analysis that use some standard security model. In many cases, however, exploitation of these vulnerabilities may not be practical and, moreover, not all application may suffer because of the identified vulnerabilities. For example, many attacks on authentication protocols[2] work because the adversary can delete some message from a sender to the receiver. In practice, deletion of a transmitted message, however, is not always practical, especially if the communication medium is wireless.

Therefore, one may need to analyze a protocol for the weaker notions of security. We introduce the notion of an *adaptable authentication model* for the type of reasoning which estimates the actual level of security provided by an authentication protocol, i.e., we infer suitable values of different parameters in the authentication model such that the protocol can be shown secure; these different parameters may include the attacker capabilities and the assumptions about the operating environment.

In the adaptable authentication model, the overall goals of an authentication protocol are broken into a finer granularity; for each fine level authentication goal, we determine the "Least Strongest-Attacker" for which the authentication goal can be satisfied. We demonstrate that this model can be used to reason about the security of supposedly insecure protocols. Such adaptability is particularly useful in those applications where one may need to trade-off the security relaxations against the resource requirements.

[1] Informally, such an attacker runs the communication network, and therefore can insert, delete, modify, delay and replay the messages. The attacker, however, can not break standard cryptographic schemes, such as encryption and signature.

[2] For example, see Chapter 3 in the book of Boyd and Mathuria [8].

Note that our methodology is fundamentally different from the traditional one in which the parameters of a security model are fixed. For example, it is not clear how to use a formal analysis tool [7] if, e.g., we assume that the attacker can not delete messages, without undergoing the redesign of the tool itself.

The adaptable authentication model makes it possible to reason about the weaker notions of security (such as if an attacker can not delete messages); not doing so may result in using inefficient protocols and discarding slightly insecure protocols. Not all real-world applications require the highest level of security (e.g., see Ksiezopolski et al.[16]); and some of the rest can not afford to implement the required level of security due to resource constraints (e.g., see Burmester et al.[9], Lindskog's thesis [17]).

In practice, most of the cryptographic protocols are designed to achieved additional security goals, besides entity authentication, because a subsequent communication is required. Some of these goals are defined in the following.

Freshness If an entity A verifies that a value v is currently sent by a peer entity then A is said to achieve the goal *freshness* for v.

Integrity If an entity A verifies that a value v is same as sent by a peer entity (who may be unknown) then A is said to achieve the goal *integrity* for v.

Authenticity If an entity A verifies that a value v is sent by some X, such that $X \in \Delta$, then A is said to achieve Δ-authenticity for v.

Traceability The goal *traceability* for an entity A is achieved by an adversary if it is possible to extract the identity of A from the protocol.

Linkability The goal *linkability* is achieved by an adversary if two instances a protocol, which involve the same initiator or responder, are linkable.

Revelation The goal *revelation* for a value v is achieved if v can be efficiently extracted after the execution of a protocol.

Entity authentication only represents a subset of all communication security goals, still there are many applications where mere authentication — without subsequent communication — is required. One such domain, which we also consider in this report, is RFID. One motivation of doing so is to convey the core idea behind the *adaptable authentication model* in its simplest form. Nevertheless, we believe that the core idea can be extended to other communication security goals[3].

The design of RFID based security protocols is a relatively challenging task because RFID tags are resource constrained devices with only a little memory

[3]This, however, may require a different construction of security property β that is described in the next chapter.

and limited computational and communication capabilities. These constrains mandate the use of light-weight cryptography. This is not the only challenge however. The generic attacker for typical RFID system is very powerful, as the tags are not assumed to be tamper-proof. Moreover, privacy also needs to be considered in the design of RFID protocols — the use of RFID based identification in wearable items can lead to serious *privacy* concerns, e.g., some people may not want to be publicly identifiable if they wear clothes that have embedded RFID tags.

We believe that *Adaptable authentication model* could be extremely useful in such applications where it is not feasible to implement the highest level of security and where the need of a trade-off, between security and resource constraints, exists. We however do not consider any of RFID domain specific problems, e.g., the efficiency of reader side of a protocol, side channel attacks and relay attacks, which may also be crucial to feasibility of an authentication protocol.

We start by briefly considering the related work in § 1.1. In Chapter 2 we describe the theoretical foundation of the model. This is followed by the concrete definitions of the model in § 2.2. We consider a simple RFID system and reason about the security of a generic authentication protocol in Chapter 3. In Chapter 4 we discuss the usefulness of our contribution and at last, in § 4.1, we conclude the work. All the proofs of security can be found in Appendix A.

1.1 Related Work

Here we only mention some of the work directly related to flexibility in security models. Various other style of formal definitions for authentication can be found, e.g., spi-calculus [2], type theory [1]. The readers are referred to Avoine's thesis [5] for a general introduction and prior art of RFID security and privacy at that time. Outside the RFID community, the book of Boyd et al. [8] contains a large number of authentication protocols and the list of reported attacks against them.

Security and performance trade-offs in client-server environments are studied in Authenticast [23], which is a dynamic authentication protocol. The adaptation is due to flexible selection of key length, algorithm and the percentage of total packets that are authenticated. The term Quality of Protection (QoP) is also used to describe adaptable security models. Ong, et al.[20], address the problems introduced by the traditional view of security — a system is either secure or insecure— by defining different security levels based on key size, block size, type of data and interval of security.

Hager [15], in his thesis, consider the trade-offs of security protocols in wireless network; but, the security adaption is only on the basis of performance, energy, and resource consumption. Covington, et al.[11], propose Pa-

rameterized Authentication, in which quality of authentication is described in terms of sensor trustworthiness and the accuracy of the measurements.

Lindskog [17] develop some solutions in his thesis, for tuning security for networked applications. The proposed methods are however limited to confidentiality. Instead of using just one instance of authentication protocol, in some approaches, e.g., by Ganger [14], over a period of time a system can fuse observations about the entities into a kind of probabilistic authentication.

Ksiezopolski, et al.[16], describe the problem of an unnecessarily high level of security that may have impacts on dependability; they present an informal model of adaptable security, which, however, is difficult to justify for the soundness of results. Sun, et al.[25], propose an evaluation method for QoP, based on normalized weighted tree. Most of the existing QoP based approaches can not be justified on concrete basis of modern cryptography. Interestingly, most of the foundational work for adaptable authentication is in the domain of RFID.

Many of the proposed RFID protocols, [6][18], are too heavy for low cost tags, and not supported by EPCGen2 [24]. Burmester, et al.[9], reports that five different proposals that are compliant to EPCGen2 but have some security vulnerabilities. Currently, there are many parallel efforts going on to develop (adaptable) privacy models that can be used to capture the security and privacy requirements in order to optimize the resource requirement for the protocols.

Damgård et al.[12] study the trade-offs between complexity and security using secret key cryptography. They propose a weaker but more practical notion of privacy; strong privacy requires a separate and independent key for each of the RFID tags. Vaudenay [26] uses eight different attacker models to reason about the privacy of RFID identification protocols. The strongest notion of privacy in Vaudenay's model is shown to be impossible to achieve; even the two other strong models mandate the use of public key cryptography. This model also serve as a inspiration for much of the following work where the authors model the adversary as a class of attacker models, e.g., Paise et al.[21], Yu Ng et al. [19] and Canard et al.,[10].

The proposal in this report is radical in the sense that we define adaptability over the definition of authentication. In all previous work, adaptability is defined over the attacker's capabilities, strength of cryptographic algorithms, trustworthiness of credentials or use of multiple channels (e.g., multi-factor authentication, context-aware authentication). Some of our previous work, [4][3], can be considered as a pre-cursor to this work; the major developments in this report are the proposal of general operational definitions for authentication, use of probabilistic proofs for LSAs (least strongest attackers) and the evaluation of our model in the RFID domain.

Chapter 2

The Adaptable Model

Traditionally, protocols are analyzed for security using a fixed security model that typically incorporates an all powerful attacker. The end-result of such an analysis is the answer to whether or not the given protocol is secure or not. This approach is intuitively described in the following metaphoric example.

Imagine a system where the only feasible operation is multiplication and our task is to establish the truth value of expressions of type $a.x \geq b$, where $a, b, x \in D$ and D is a finite subset of natural numbers. Let, a and b are the given values in the specification of expression, while x is an unknown value. Of course, the natural approach for solving this problem is to assume $x = x^{\perp}$, where $x^{\perp}$ is the smallest number in D and then evaluate the expression, $a.x^{\perp} \geq b$. This approach is similar to how security is analyzed traditionally — in which the evaluation of the expression is replaced by the security analysis, a corresponds to a given protocol, b is the required security goals for a, and $x^{\perp}$ is the security model that incorporates the all powerful attacker.

On the other hand, by initially assuming a given protocol as a secure protocol, we may need to find out the "strongest" security model for which the initial assumption remains valid. Metaphorically, we assume $a.x \geq b$ to be true and try to find out the smallest x in D such that this assumption remains valid. For this purpose, a naive (brute-force) approach is to iterate the evaluation of expression: start with the largest x; if the expression is valid then decrement x and repeat the process, otherwise, the value of x in previous iteration is the solution. Similarly, such a naive approach for security requires iterating the security analysis over the possible types of security models.

In our proposal, we present a single framework that can be used to find out such a "strongest security model". Metaphorically, we try to solve $a.x \geq b$ as an inverse problem, even though $x = a/b$ is not feasible to compute directly. The precise description of our main idea and its formal construction is in the following.

2.1 Theory

Let Θ be a background theory[1], α be an attacker model, Π be an authentication protocol and G be one of the authentication goals[2]. The standard approach towards security analysis is in the following general form.

$$\Theta, \Pi, \alpha \models G \tag{2.1}$$

The above argument represents the security analysis as a process in which one tries to show, under the security model $\{\Theta, \alpha\}$, that G can be achieved by running an instance of Π. It must be noted that the above argument is not confined to classic mathematical logic; the meaning of '$\models$' depends on the type of analysis, e.g., in complexity theoretic cryptography $\models$ stands for reductionist type proofs, in formal analysis it may stand for static analysis [2].

We aim to formulate the above argument as an inverse problem, in order to determine certain parameters of the security model such that G can be inferred. In place of Equation 2.1, we use the following two abductive style arguments in the Adaptable authentication model.

$$\Theta, \Pi, \beta \models G \tag{2.2}$$

$$\Theta, \Pi, \alpha \models \beta \tag{2.3}$$

We refer to Equation 2.2 as the *Beta argument* and Equation 2.3 as the *Alpha argument*. These two arguments are proposed to divide the classic authentication problem (Equation 2.1) in two, by introducing an intermediate statement β. The concrete form of β is defined in § 2.2. For now, β may be considered as a special security property over the protocol messages.

Two types of processes are involved in both of the arguments above. The first one is an *abduction process*, which refers to hypothesizing β in the Beta argument and α in the Alpha argument. The second process is called *validation process*, which deals with the validity of the arguments itself. So, in total we need to consider the following four processes.

1. Abduction process in the Beta argument, i.e., hypothesizing β
2. Validation process in the Beta argument, i.e., validating G
3. Abduction process in the Alpha argument, i.e., hypothesizing α
4. Validation process in the Alpha argument, i.e., validating β

The process of hypothesizing β is trivial, as a Π in the class of authentication protocols typically contains a relatively small number of base terms.

[1] A security model may consists of environment model, system model, attacker model α, semantics of terms, etc; Θ refers to all these components except α.

[2] For more details see Definition of Authentication [4].

So, we can exhaustively search the possible space of β. For example, if Π is a challenge response protocol then there are only two messages, a challenge m_1 and a response m_2; the possibilities for which β may be satisfied are only three: $\{m_1, m_2\}$, $\{m_1\}$ and $\{m_2\}$.

The *validation process* for G in the Beta argument is defined in § 2.2 under the names of *operational definitions*. A valid β in the Beta argument is *just an hypothesis* in the complete model and thus a separate validation argument for β is required, which is the Alpha argument. If the Alpha argument can be validated, for some α, then β becomes a valid statement, but apparently at the cost of α being a hypothesis.

Traditionally, the issue 'α as a hypothesis' — i.e., α really models the attacker for a specific domain? — is not so important because α usually models an 'all powerful attacker'. For example, the Dolev-Yao attacker [13] is believed to subsume all conceivable attackers in most computer networks. So, one can safely assume the validity of the hypothesis (Dolev-Yao attacker) while, e.g., deploying a 'secure' authentication protocol in a cooperate network. In our proposal, however, the hypothesis α may correspond to a weaker attacker. Therefore, α obtained in the *Adaptable* authentication model should not be assumed valid by default and a decision, whether α is reasonable to assume or not, must be made in an application specific manner.

For the Alpha argument, we outline a Hybrid process that incorporates both the abduction process (i.e., hypothesizing α for a given β) and a validation process (i.e., validating β using Θ, Π and α). The Hybrid process is overall sound[3] if the validation process is sound.

The Hybrid process: The Hybrid process starts with the validation process of the Alpha argument by assuming the capabilities of most powerful attacker, say α_i, relevant to the given system model. If the validation process fails then depending on the cause of failure we weaken α_i to α_{i+1} and continue with the validation process once again; the loop continues until the validation process succeed. The α for which the validation process succeed is the *least strongest attacker* (LSA).

Consequently, the resultant LSA is not necessarily correspond to some standard attacker model, such as the Dolev-Yao attacker [13]. As long as the *validity process* is sound, α_i at *ith* iteration is rejected if it is not a valid hypothesis. The existence of any undiscovered 'stronger' LSA does not invalidate the earlier results, as any weaker attacker model is always a subset of the stronger model. Thus, the soundness of the Hybrid process is not a defeasible guarantee[4], as long as the background theory Θ remains the same.

The Alpha and the Beta arguments could be probabilistic,[5] i.e., there is

[3] But, the process may not be complete or optimum.

[4] In classic reasoning, a valid but defeasible argument has the possibility of invalidation when more premises are added.

[5] For example, in complexity theoretic proofs of security (e.g., see §2.6 [8])

some probability $p < 1$ that a goal G is valid assuming the validity of Θ, Π and β. Similarly, there is some probability $q < 1$ that a β is valid assuming the validity of the premises Θ, Π and α. In the complete model, G is achieved with a probability that is a function of p and q. The probabilistic concerns do not arise until we actually start validating the alpha and beta arguments.

To sum up, we highlight the two aspects that form the basis of the adaptability of *Adaptable authentication model*. The first aspect is the way we formulated the Alpha and the Beta arguments; the Beta argument is 'attacker independent'; the Alpha argument is 'goal independent'. The second aspect is the abductive style treatment of the two arguments, namely the traditional security problem (whether G can be satisfied under a standard security model) is formulated as an inverse problem by breaking it in to four smaller problems: *abductive* and *validation* process for both Alpha and Beta arguments. This allows us to infer an actual security model for which a given authentication protocol is secure.

2.2 Binding Sequence

In this section, we present the generic forms of (1) β, (2) the validation process of the Beta argument and (3) a restricted form of the Hybrid process. The validation of the Beta argument means how to infer an authentication goal G from Θ, Π and β; in our case, however, G is one of the authentication goals. We define this validation process with the operational definitions of G that are in terms of β. We define the Hybrid process at an abstract level, i.e., for arbitrary form of α, Π and Θ; an example of β with a concrete Hybrid process (i.e., for the specific choices of α, Π and Θ) can be found in Chapter 3

Consider a network of entities who communicate with each other through message passing protocols; the protocols under consideration are of entity authentication. A protocol Π may be executed among two or more entities. An instance of Π is denoted by $\Pi(i)$, where i is an index. We propose the following definition of *binding sequence* as a concrete form of β.

Binding Sequence: A sequence of protocol messages is called a binding sequence β_X if the violation of any of the following properties generate an efficiently detectable event for an entity X.

1. Deletion/Insertion/Modification of a message in β_X.
2. Changing the sequence of messages in β_X.

Intuitively, a binding sequence is a list of selected messages that preserves their 'integrity', as all unauthorized changes in β_X should be detectable for X; of course, β_X may be replayed. Note that such an integrity property of β_X is different from the integrity of the messages it contains. A β_X can be constructed from completely unauthenticated messages.

We define the following encoding from a *list* L to a *set* S. Let the size (the number of elements) of list is $|L|$ and every element in the list is addressable by a unique index i such that $1 \leq i \leq |L|$. The ith element in L is l_i and is encoded in the following set s_i, such that $s_i \in S$: $s_i ::= \{l_i\} \cup \{l^s_{i-1}\}$. In this way, a binding sequence β_X has an equivalent set representation and therefore the following set operations are defined accordingly: $\in$, $\subset$ and $\subseteq$.

2.3 Validation process of the Beta argument

Next, we propose *operational definitions* as a validation process of the Beta argument, namely we define the authentication goals for X in terms of β_X. Our formulation is for a two-party case, but it is trivial to extend them to multi-party case[6]. In the following A and B denote specific network entities, while X represents an arbitrary network entity. These operational definitions are also illustrated in Figure 2.1.

Existence: Let $\beta_A(i)$ and $\beta_A(j)$ be generated when A executes the protocol with B and X respectively. If A can efficiently distinguish between $\beta_A(i)$ and $\beta_A(j)$ (for all choices of X) then A is said to achieve the goal *existence* of B from $\beta_A(i)$. This is abbreviated as $\text{EXST}(A \rightarrow B, \beta_A(i))$.

Operativeness: Let $\beta_A(i)$ and $\beta_A(j)$ be generated when A executes the protocol twice with X . If A can efficiently distinguish between $\beta_A(i)$ and $\beta_A(j)$ (for all choices of X) then A is said to achieve the goal *operativeness* for X. This is abbreviated as $\text{OPER}(A \rightarrow B, \beta_A(i))$.

Willingness: Let $\beta_B(i)$ and $\beta_B(j)$ be generated when A and X execute the protocol with B. If B can efficiently distinguish between $\beta_B(i)$ and $\beta_B(j)$ for all choices of X then A is said to achieve the goal *willingness* for B, from the corresponding binding sequence $\beta_A(i)$. This is abbreviated as $\text{WLNG}(A \rightarrow B, \beta_A(i))$.

Single-sided Authentication: If an entity A achieves $\text{OPER}(A \rightarrow B, \beta_A(i))$, $\text{EXST}(A \rightarrow B, \beta_A(i))$, and $\text{WLNG}(A \rightarrow B, \beta_A(i))$ then A is said to achieve single sided authentication, abbreviated as $\text{SATH}(A \rightarrow B, \beta_A(i))$.

In the above definition, the goals — existence, operativeness and willingness —, are achieved in the same instance $\beta_A(i)$. If we assume that an entity never proceeds from a step in a protocol run if she does not receives the expected message in that step then we have the following definition of confirmation.

[6] For instance, if A is supposed to achieve G for both B and C then the operational definition of G for A to B and for A to C must be satisfiable using a common β_X.

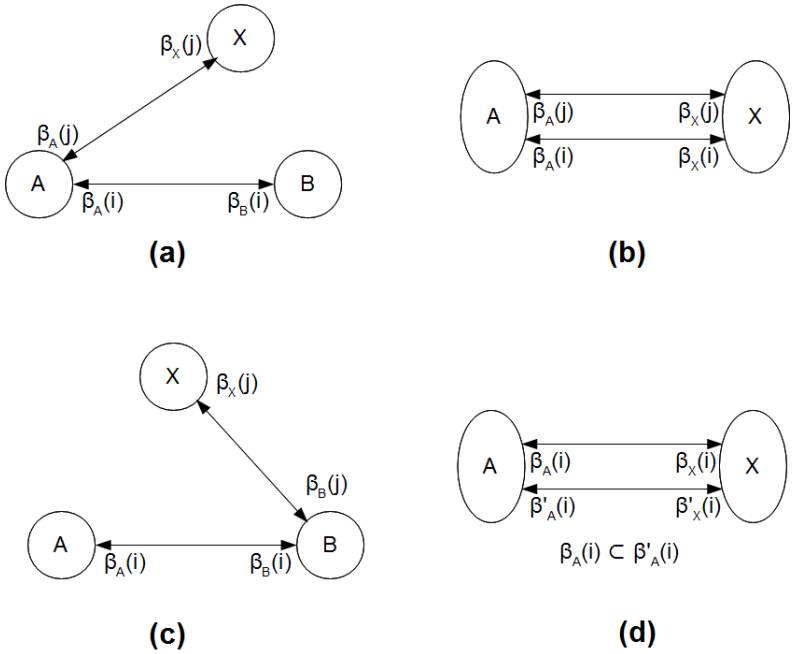

Figure 2.1: Operational Definitions: (a) Existence (b) Operativeness (c) Willingness (d) Confirmation

Confirmation: Let $\beta_A(i)$ and $\beta'_A(i)$ be generated in a single run when A executes the protocol with X and $\beta_A(i) \subset \beta'_A(i)$. If A achieve some goal G from $\beta_A(i)$ then A has the confirmation on G in $\beta_A(i)$. This is abbreviated as CNFM$(A \rightarrow B, \beta_A(i))$.

Intuitively, a subsequent message that is part of a binding sequence carries the assurance that the peer entity has accepted all previous messages.

Strong Single-sided Authentication: If an entity A achieves SATH$(A \rightarrow B, \beta_A(i))$ and CNFM$(A \rightarrow B, \beta_A(i))$ then A is said to achieve *strong single-sided authentication*, abbreviated as SSATH$(A \rightarrow B, \beta_A(i))$.

Intuitively, the above definition introduces the requirement of a "confirmation message". Therefore, now, A has the assurance that B knows about the achieved goal that A has authenticated B.

Mutual Authentication If an entity A achieves SATH$(A \rightarrow B, \beta'_A(i))$ and CNFM$(A \rightarrow B, \beta_A(i))$, such that $\beta_A(i) \subset \beta'_A(i)$, $\beta_A(i) \models \beta_B(i)$ and SATH$(B \rightarrow A, \beta_B(i))$, then A is said to achieve *mutual authentication*, abbreviated as MATH$(A \rightarrow B, \beta_A(i))$.

Strong Mutual Authentication If an entity A achieves $\text{SATH}(A \rightarrow B, \beta_A(i))$ and $\text{CNFM}(A \rightarrow B, \beta_A(i))$, such that $\beta_A(i) \models \beta_B(i)$ and $\text{SATH}(B \rightarrow A, \beta_B(i))$, then A is said to achieve *strong mutual authentication*, abbreviated as $\text{SMATH}(A \rightarrow B, \beta_A(i))$.

The difference between the above two definitions is in the relaxation, $\beta_A(i) \subset \beta'_A(i)$. Due to this relaxation A may not have the assurance that B knows that A has authenticated B. In the case of *strong mutual authentication*, A do have this assurance, as the confirmation is on the same binding sequence that is used in the single-sided authentication.

Next, we turn to the privacy related goals. We introduce a notion of *controllability* as follows. A Π is *controllable* by peer entities if the union of their binding sequences, $\bigcup_Y \beta_Y$, contains all the messages of Π, where Y is some peer entity in Π. So, controllability implies that each protocol message is part of at least one binding sequence belonging to the peer entities. Consequently, we have the following relation for an instance $\Pi(i)$ of a *controllable* protocol: $\Pi(i) = \bigcup_Y \beta_Y(i)$.

Anonymity Let $\Pi(m)$ and $\Pi(n)$ be the two instances of a controllable Π, such that $\Pi(m)$ involves an entity X and $\Pi(n)$ does not involve X, as an initiator or responder. The goal *Anonymity*, abbreviated as $\text{ANMT}(X, \Pi)$, is achieved if an adversary A can not distinguish between $\Pi(m)$ and $\Pi(n)$.

Untraceability Let $\Pi(m)$, $\Pi(n)$ and $\Pi(o)$ be the three instances of a controllable Π, such that $\Pi(m)$ and $\Pi(n)$ involve X as an initiator or responder, while $\Pi(o)$ does not involve X. The gaol *untraceability*, or $\text{UNTC}(X, \Pi)$, is achieved if an adversary A can not pick the pair $(\Pi(m), \Pi(n))$ with significantly different probability than either of the rest of two pairs, namely $(\Pi(m), \Pi(o))$ and $(\Pi(n), \Pi(o))$.

2.4 Abstract Hybrid Process

At last, we define an abstract form of the Hybrid process, which is used to hypothesize α and to validate β at the same time. For simplicity, we restrict the abstract form to those protocols in which the number of messages are fixed. Moreover, we assume that all messages are well typed. With these restrictions, the only property that needs to be checked (as per definition of binding sequence) is *modification*.

Consider an arbitrary binding sequence: $\beta_X = [m_1, ..., m_n]$, where $n \geq 1$. Let m_i, with $i < n$, be an *unmodified* message in β_X. Let m'_i represents a message obtained by modifying m_i. Let $\text{ACCEPT}(\beta_X)$ be an event that X accepts[7] β_X. Let M_{β_X} be a set that contains all choices of modified β_X, e.g.,

[7] X accepts β_X as per the definition of binding sequence.

if $\beta_A = [m_1, m_2]$ then $M_{\beta_A} = \{[m'_1, m_2], [m_1, m'_2], [m'_1, m'_2]\}$. In general, M_{β_X} contains $2^n - 1$ elements.

The generic Hybrid process: We consider each element of M_{β_X} and calculate $Pr(\text{ACCEPT}(\beta^i_X))$, where $\beta^i_X \in M_{\beta_X}$ and $1 \geq i \leq (2^n - 1)$. If $Pr(\text{ACCEPT}(\beta^i_X))$ is negligible then we consider another element in M_{β_X}, until no element is left in M_{β_X}. If $Pr(\text{ACCEPT}(\beta^i_X))$ is not negligible, we include reasonable assumptions so that $Pr(\text{ACCEPT}(\beta^i_X))$ is negligible under new assumptions. These additional assumption could be related to α or the environment in which Π operates. In the worst case, α is empty.[8]

In the next chapter, we demonstrate the utility of the Adaptable authentication model by applying the approach to a relatively simple, but realistic, system, namely RFID authentication. The purpose of the report, however, is not to address the general RFID authentication and privacy problems.

[8]This corresponds to an attacker with no capabilities besides what is assumed in the ideal execution of protocol.

Chapter 3

Case Study: A Simple RFID System

All RFID tags that we consider here are passive transponders identified by an ID; the ID is not necessarily unique. In reality this ID may correspond to the item to which the tag is attached.

We define four classes of attacker capabilities: Destructive, α^D; Forward, α^F; Weak, α^W; and Coward, α^C. An attacker A may belong to one of these classes, which are modeled over the following set of oracles.

- CreateTag(k): A can create a tag with a key k stored in it.
- Launch($\Pi(i)$,ID): A can interact with a tag *ID* of his choice.
- Respond($\Pi(i)$): A can respond to a RFID reader's interrogation.
- Corrupt(ID): A can read all memory contents of any tag *ID*.

A in Destructive class can access all of these oracles, but, if A access Corrupt oracle then the tag ID is destroyed as the tampering is assumed to be detectable. Forward class is same as Destructive one, except no other type of oracle can be accessed after an access to Corrupt oracle is made; additional Corrupt queries are allowed. A in Weak class can not access Corrupt oracle. A in Coward class can not invoke Corrupt and Respond oracles.

We consider a relatively simple system model, where there is one reader R and n number of RFID tags[1]: ID_i, $1 \leq i \leq n$; each tag is attached to an item in *Warehouse*. There is a single entry point *Entry* into Warehouse and a single exit point *Exit* from Warehouse. Whenever an item passes through Entry, it gets attached with a tag ID_i and a relevant entry (ID_i, K_i) is stored in the reader's database. When an item arrives at Exit, the tag is identified as ID_i

[1] The assumption of single reader is ubiquitous in most of the literature, e.g., see Damgård [12], Vaudenay [26].

by executing a protocol that is derived from the generic RFID authentication protocol, which we specify later. When a tag ID_i leaves Exit, it is killed (i.e., no more radio communication is possible with ID_i) by R and database entry (ID_i, K_i) is removed.

The reader R may have multiple front-ends at Entry and Exit but has one common back-end. In reality, the back-end could be a distributed system in which the database of tags is synchronized in real-time. Whenever a tag is in Exit, the authentication protocol is automatically executed.

An attacker A, in some attacker's class, exists inside the warehouse from where he can invoke the oracle queries following the rules of the attacker class. In addition to that, A also has access to customer records. Privacy is defined as protecting the relation between the item bought and the customer. We assume that an attacker is not capable to 'physically observe' the item all the way to a customer, but, if an attacker can link communication of a tag ID to a customer at Exit then he can violate the privacy.

Usually, the concrete definitions of *privacy* and *security* are system dependent. In our case, *anonymity* is not a concern as attacker is already in Warehouse. Similarly, the reader at Exit is assumed honest and therefore *willingness* is not a concern for security. We use the following definitions for our RFID system.

Privacy: A protocol Π, involving R as an initiator and ID as a responder, is *private* if untraceability, $\mathrm{UNTC}(\mathrm{ID}, \Pi)$, can be achieved.

Security: A protocol Π , involving R as an initiator and ID as a responder, is *secure* if existence, $\mathrm{EXST}(R \rightarrow \mathrm{ID}, \beta_R)$, and operativeness, $\mathrm{OPER}(R \rightarrow \mathrm{ID}, \beta_R)$, can be achieved.

We consider the following generic RFID protocol, Π. Note that many of the existing protocols follow this generic form, e.g., weak-private RFID schemes [26].

$1.R \rightarrow \mathrm{ID} : \mathrm{challenge} = V_R$
$2.\mathrm{ID} \rightarrow R : \mathrm{response} = F_K(V_R, V_{\mathrm{ID}})$

In this generic form, $F_K(...)$ is a PRF (pseudo-random function), V_R is the value generated by the reader and V_{ID} is the value generated by the tag. The reader R can find (ID,K) in the database by the searching the value of K from database, such that the following predicate is true: $p(\mathrm{challenge}, \mathrm{response}, K, x)$. For instance, in Π_3 (Appendix A.3) we have, $p(N_R, , \mathrm{response}, k, x = \phi) ::= (\mathrm{response} = F_k(N_R, b))?$ true : false.

We can construct eight different concrete protocols, say $\Pi_1, \Pi_1, ..., \Pi_8$, over the following three parameters: the challenge V_R is random or not; the value V_{ID} is random or not; and the key K is different or not for each tag.

Next, we analyze the eight concrete protocols for security and privacy, using the Adaptable authentication model. As we may recall, there are three processes involved in the analysis, i.e., *abduction* and *validation* process for Beta argument and the Hybrid process for Alpha argument. The *abduction* process is trivial, namely $\beta_X = [V_R, F_K(V_R, V_{\mathrm{ID}})]$ for the generic protocol. The *validation* process corroborates the *operational definitions* of relevant Gs, namely $\mathrm{EXST}(R \rightarrow \mathrm{ID}, \beta_R)$ and $\mathrm{OPER}(R \rightarrow \mathrm{ID}, \beta_R)$ for the *security*, and $\mathrm{UNTC}(\mathrm{ID}, \Pi_i)$ (where $1 \leq i \leq 8$) for the *privacy*.

A concrete Hybrid process can be derived from the generic Hybrid process described in §2.2. In all of the protocols, *deletion*, *insertion* and *reordering* is not possible, which justifies the restrictions of the abstract Hybrid process. So, we only analyze for any undetectable modifications in β_R. Let A be an attacker from one of the attack classes. Let $\mathrm{ACCEPT}(\beta_R)$ be an event that the reader accepts β_R. The generic, but concrete, Hybrid process is as follows.

1. $Pr(\mathrm{ACCEPT}([W'_R, F_K(W_R, W_{\mathrm{ID}})]))$:
   This case analyzes the probability of the event that A select W'_R, where $W'_R \neq W_R$, in such a way that $F_K(W_R, W_{\mathrm{ID}})$ can be returned to the reader.

2. $Pr(\mathrm{ACCEPT}([W_R, F_K(W_R, W_{\mathrm{ID}})']))$:
   This case analyzes the probability of the event that A compute the response $F_K(W_R, W_{\mathrm{ID}})'$, where $F_K(W_R, W_{\mathrm{ID}})' \neq F_K(W_R, W_{\mathrm{ID}})$, in such a way that probability of ACCEPT is close to 1.

3. $Pr(\mathrm{ACCEPT}([W'_R, F_K(W_R, W_{\mathrm{ID}})']))$:
   This case analyzes the probability of the event that A select W'_R, where $W'_R \neq W_R$, and compute the response $F_K(W_R, W_{\mathrm{ID}})'$, where $F_K(W_R, W_{\mathrm{ID}})' \neq F_K(W_R, W_{\mathrm{ID}})$, in such a way that probability of ACCEPT is close to 1.

The actual security proofs for the eight concrete protocols can be found in Appendix A; the result of the analysis is summarized in Table 3.1. Each row in the table corresponds to one of the concrete protocols; the specific choices made for W_R, W_{ID} and K are mentioned in the corresponding columns. The last column summarizes the result of our analysis, in form of assumptions required to justify the security and privacy of these protocols. The details of these assumptions are presented in the following list.

(a) The *existence* is not achieved, so, e.g., same type of items should be in Warehouse, or there should be some auxiliary (e.g., physical) mechanism to distinguish between items if they are different.
(b) Only one item is presented to the reader R at a time.
(c) The *operativeness* is not achieved, so, e.g., visual inspection should be done at Exit to make sure the item with ID is currently there.

Protocol	K is different	V_{ID} is random	V_R is random	Results
Π_1	No	No	No	{a,b,c,Destructive}
Π_2	No	No	Yes	{a,b,Weak}
Π_3	No	Yes	No	{a,b,d,e,Coward}
Π_4	No	Yes	Yes	{a,b,e,Coward}
Π_5	Yes	No	No	{c,h,Destructive }
Π_6	Yes	No	Yes	{Destructive }
Π_7	Yes	Yes	No	{d,e,Destructive}
Π_8	Yes	Yes	Yes	{e,Destructive}

Table 3.1: Concrete Forms of the Generic Protocol

(d) The reader R should query a tag more than once to detect collisions in the values of W_{ID}.

(e) $F_k(...)$ is a PRP (pseudo-random permutation) with an efficiently computable inverse function $F_K^{-1}(...)$ (e.g., AES encryption).

(h) Privacy is not possible to achieve within the model, so, privacy should not be a concern for the items in Warehouse.

Let us consider, for instance, the case Π_2 in the table. The protocol Π_2 corresponds to a system where all RFID tags share a common key and there is no pseudo-random generator implemented on the tags. The i^{th} interrogation of the reader consists of a random challenge $N_R(i)$. As shown in the last column that the protocol is secure and private against Weak class of attackers, as long as the assumption, (a) and (b), are satisfied. This example illustrates the types of results that we obtain in Adaptable authentication model, i.e., the appropriate parameters of authentication model required to justify security and privacy.

Chapter 4

Discussion

In traditional security analysis, most of the concrete protocols in Table 3.1 are 'insecure', with the possible exceptions of Π_6 and Π_8 (e.g., see the case of strong privacy [12]). This is due to a strict deductive style interpretation of classic security argument (Eq. 2.1) and use of a strict notion of attacker in the corresponding security model. For example, if V_R is not random in a concrete protocol (i.e., one of Π_1, Π_3 or Π_5) then the protocol is simply considered 'insecure' under Dolev-Yao attacker [13] — as it is prone to replay attack.

On the other hand, we consider security and privacy as an a-priori assumption, even for the weakest protocol Π_1. We infer an attacker model and a set of assumptions about the environment such that the security can be justified. It is not always possible to justify a security goal within the model, but this does not necessarily mean a dead-end to the analysis; Since our security and privacy goals are formulated at primitive level (compared to single high level formulation, e.g., matching conversation [8]), impossibility of one fine level goal does not imply such impossibility for all other goals.

In long run, the proposed model needs to be evaluated on a broader scale, with more authentication goals and for more general RFID system, albeit, the results in Table 3.1 are promising for the simple RFID system. In particular, there are two open questions that we like to address in future. First, we need to determine whether or not the validity process of β , namely the Alpha argument, is at most as hard as that of standard security analysis (Eq. 2.1). Secondly, whether or not the operational goals in terms of binding sequences are equivalent to some of other standard formulation, e.g., matching conversation [8], Blinders [26].

4.1 Conclusion

Adaptable authentication model can be used to reason about authentication protocols that are not normally considered secure as per 'standard authentication model'. We propose the Adaptable authentication model as a step

towards exploring a rather unexplored area of weaker security. The results are particularly useful when one needs to optimize security for resource constraints, e.g., the protocols Π_1 and Π_2, which only require a hash function on the tags, may suffice for the requirements of certain applications.

It must be noted that the Adaptable authentication model does not change the actual security of a protocol; the model just capture the weak form of security which otherwise labeled as insecurity under standard model. System designers, therefore, must be cautious while interpreting the results obtained in the Adaptable model; security guarantees are accompanied by extra assumptions and typically a weaker attacker model, which may not be justifiable for every conceivable environment.

Bibliography

[1] Martín Abadi. Secrecy by typing in security protocols. *J. ACM*, 46:749–786, September 1999.

[2] Martín Abadi and Andrew Gordon. Reasoning about cryptographic protocols in the spi calculus. In *CONCUR '97: Concurrency Theory*, volume 1243 of *Lecture Notes in Computer Science*, pages 59–73. Springer Berlin / Heidelberg, 1997.

[3] Naveed Ahmed and Christian D. Jensen. Entity authentication:analysis using structured intuition. In *NODES-10: 4th Nordic Workshop on Dependability and Security*. IMM, DTU, Denmark, 2010.

[4] Naveed Ahmed and Christian Damgaard Jensen. Definition of entity authentication. In *Security and Communication Networks (IWSCN), 2010 2nd International Workshop on*, pages 1 –7, May 2010.

[5] G. Avoine. Cryptography in radio frequency identification and fair exchange protocols. *PhD thesis*, 2005.

[6] G. Avoine and P. Oechslin. A scalable and provably secure hash-based rfid protocol. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 110 – 114, March 2005.

[7] C. Bodei, M. Buchholtz, P. Degano, F. Nielson, and H. Riis Nielson. Automatic validation of protocol narration. In *Computer Security Foundations Workshop, 2003. Proceedings. 16th IEEE*, pages 126–140, july 2003.

[8] C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment*. Springer Book, 2003.

[9] M. Burmester and J. Munilla. *A Flyweight RFID Authentication Protocol*. Eprint: IACR, `http://eprint.iacr.org/2009/212`, 2009.

[10] S. Canard, I. Coisel, J. Etrog, and M. Girault. *Privacy-Preserving RFID Systems: Model and Constructions*. Eprint: IACR, 2010.

[11] Michael J. Covington, Mustaque Ahamad, Irfan Essa, and H. Venkateswaran. Parameterized authentication. In Pierangela Samarati, Peter Ryan, Dieter Gollmann, and Refik Molva, editors, *Computer Security — ESORICS 2004*, volume 3193 of *Lecture Notes in Computer Science*, pages 276–292. Springer Berlin / Heidelberg, 2004.

[12] Ivan Damgård and Michael Østergaard Pedersen. Rfid security: tradeoffs between security and efficiency. In *Proceedings of the 2008 The Cryptopgraphers' Track at the RSA conference on Topics in cryptology*, CT-RSA'08, pages 318–332, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198 – 208, March 1983.

[14] G. R. Ganger. Authentication confidences. *CMU-CS-01-123, Technical Report*, 2001.

[15] C. T. Hager. Context aware and adaptive security for wireless networks. *PhD thesis*, 2004.

[16] Bogdan Ksiezopolski and Zbigniew Kotulski. Adaptable security mechanism for dynamic environments. *Computers & Security*, 26(3):246 – 255, 2007.

[17] S. Lindskog. Modeling and tuning security from a quality of service perspective. *PhD thesis*, 2005.

[18] David Molnar, Andrea Soppera, and David Wagner. A scalable, delegatable pseudonym protocol enabling ownership transfer of rfid tags. In Bart Preneel and Stafford Tavares, editors, *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 276–290. Springer Berlin / Heidelberg, 2006.

[19] Ching Ng, Willy Susilo, Yi Mu, and Rei Safavi-Naini. Rfid privacy models revisited. In Sushil Jajodia and Javier Lopez, editors, *Computer Security - ESORICS 2008*, volume 5283 of *Lecture Notes in Computer Science*, pages 251–266. Springer Berlin / Heidelberg, 2008.

[20] Chui Sian Ong, K. Nahrstedt, and Wanghong Yuan. Quality of protection for mobile multimedia applications. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 2, pages II – 137–40 vol.2, July 2003.

[21] Radu-Ioan Paise and Serge Vaudenay. Mutual authentication in rfid: security and privacy. In *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, ASIACCS '08, pages 292–299, New York, NY, USA, 2008. ACM.

[22] A.W. Roscoe. Intensional specifications of security protocols. In *Computer Security Foundations Workshop, 1996. Proceedings., 9th IEEE*, page 28 38, June 1996.

[23] P.A. Schneck and K. Schwan. Dynamic authentication for high-performance networked applications. In *Quality of Service, 1998. (IWQoS 98) 1998 Sixth International Workshop on*, pages 127 –136, May 1998.

[24] EPC Global Ratified Specification. *EPCglobal Tag Data Standards Version 1.3*. `http://www.epcglobalus.org`, March 8, 2006.

[25] Yan Sun and A. Kumar. Quality-of-protection (qop): A quantitative methodology to grade security services. In *Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on*, pages 394 –399, June 2008.

[26] Serge Vaudenay. On privacy models for rfid. In *Proceedings of the Advances in Crypotology 13th international conference on Theory and application of cryptology and information security*, ASIACRYPT'07, pages 68–87, Berlin, Heidelberg, 2007. Springer-Verlag.

[27] T.Y.C. Woo and S.S. Lam. A semantic model for authentication protocols. In *Research in Security and Privacy, 1993. Proceedings., 1993 IEEE Computer Society Symposium on*, pages 178–194, May 1993.

Appendix A

Proofs of Security

In all of the following proofs a, b and k represent constant values. All nonces are unpredictable; N_R and N_{ID} are reader-side and tag side nonces respectively. K is a key variable; K_i is the key stored in tag ID_i. The values in a particular instance of a protocol are indexed, e.g., $N_{\text{ID}}(i)$ and $N_R(i)$. The generic form of hypothesis β_R is $[V_R, F_K(V_R, V_{\text{ID}})]$.

S is the size (in bits) of output of $F_K(...)$. The sizes of nonces are $|N_R|$ and $|N_{\text{ID}}|$ and $|N_R| = |N_{\text{ID}}|$. Let q be the total number of queries to $F_K(...)$ or pseudo-random generator. Let n be total number of tags in the system. Let $q < q_{th}$ and $n < n_{th}$, for suitable choices for n_{th} and q_{th} (e.g., the birthday bound).

For simplicity, we rely on asymptotic analysis for calculating different probabilities. The probabilities, $q.2^{-S}$, $q.2^{-|N_R|}$, $q.2^{-|N\text{ID}|}$ and $n.2^{-|K|}$ are assumed to be negligible. In practice this can be achieved by suitable choices of cryptographic primitives.

A.1 Π_1: (1) $R \rightarrow \mathbf{ID}_i : a$ (2) $tag_i \rightarrow R : F_k(a, b)$

First, we consider the Beta argument: $\beta_R = [a, F_k(a, b)]$. We need to show the following.

$$\Theta, \Pi_1, [a, F_k(a, b)] \models \text{EXST}(R \rightarrow \text{ID}, \beta_R(i)), \tag{A.1}$$

$$\Theta, \Pi_1, [a, F_k(a, b)] \models \text{OPER}(R \rightarrow \text{ID}, \beta_R(i)), \tag{A.2}$$

$$\Theta, \Pi_1, [a, F_k(a, b)] \models \text{UNTC}(\text{ID}, \Pi_1) \tag{A.3}$$

As per operational definition of $\text{EXST}(R \rightarrow \text{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [a, F_k(a, b)]$ (for ID) and $\beta_R(j) = [a, F_k(a, b)]$ (for arbitrary X). As $\beta_R(i) = \beta_R(j)$, the distinguishing is not possible. So, we include the assumptions, (a) and (b).

For $\text{OPER}(R \rightarrow \text{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [a, F_k(a, b)]$ and $\beta_R(j) = [a, F_k(a, b)]$, which are two different in-

stance between R and ID. Again, this is not possible even in honest model, as $\beta_R(i) = \beta_R(j)$. So, we include the assumption (c).

For UNTC($\Pi_1 \rightarrow tag_i$), adversary A should be able to pick a pair with a probability different from the rest of two pair. The three instances of binding sequence are same. Hence, Π_1 is private.

We use the Alpha argument to validate $[a, F_k(a,b)]$. Let A be in Destructive class.

1. $Pr(\text{ACCEPT}([a', F_k(a,b)]))$:
   If $a' \neq a$, the tag will not respond assuming the tag checks that the challenge is a. If the tag does not have this check then the probability of response being $F_k(a,b)$ is only 2^{-S}.

2. $Pr(\text{ACCEPT}([a, F_k(a,b)']))$:
   The event ACCEPT can only occur if $F_k(a,b)' = F_k(a,b)$, which is not possible as $F_k(a,b)'$ needs to be a modified version of $F_k(a,b)$.

3. $Pr(\text{ACCEPT}([a', F_k(a,b)']))$:
   Similarly, ACCEPT can only occur if $F_k(a,b)' = F_k(a,b)$.

Thus, A is from Destructive class.

A.2 Π_2: (1) $R \rightarrow \mathbf{ID}_i : N_R$ (2) $\mathbf{ID}_i \rightarrow R : F_k(N_R, b)$

First we consider the Beta argument. β_R in this case is $[N_R, F_k(N_R, b)]$

$$\Theta, \Pi_2, [N_R, F_k(N_R,b)] \models \text{EXST}(R \rightarrow \text{ID}, \beta_R(i)), \tag{A.4}$$

$$\Theta, \Pi_2, [N_R, F_k(N_R,b)] \models \text{OPER}(R \rightarrow \text{ID}, \beta_R(i)), \tag{A.5}$$

$$\Theta, \Pi_2, [N_R, F_k(N_R,b)] \models \text{UNTC}(\text{ID}, \Pi_2) \tag{A.6}$$

For EXST($R \rightarrow$ ID, $\beta_R(i)$), R should be able to distinguish between $\beta_R(i) = [N_R(i), F_k(N_R(i), b)]$ and $\beta_R(j) = [N_R(j), F_k(N_R(j), b)]$. Such a distinguishing is not possible as response from all tags is of same type. So, we include the assumption, (a) and (b).

For OPER($R \rightarrow$ ID, $\beta_R(i)$), R should be able to distinguish between $\beta_R(i) = [N_R(i), F_k(N_R(i), b)]$ and $\beta_R(j) = [N_R(j), F_k(N_R(i), b)]$. Such a distinguishing is efficiently computable, although, there are two situations when this is not possible: $N_R(i) = N_R(j)$ and $F_k(N_R(i), b) = F_k(N_R(j), b)$. Since N_R is the output of pseudo-random generator, the former case can only happen with negligible probability,$q.2^{-|N_R|}$. The later case can only occur with probability of $q.2^{-S}$.

For UNTC(ID, Π_2), three instances of binding sequence are as follow: $\beta_R(i) = [N_R(i), F_k(N_R(i), b)]$ (between R and ID$_i$), $\beta_R(j) = [N_R(j), F_k(N_R(j), b)]$ (between R and ID$_i$) and $\beta_R(k) = [N_R(k), F_k(N_R(k), b)]$ (between R and ID$_j$).

This is not possible as all tags have the same key k and there is no difference between instances of different tags. Thus, Π_2 is private.

Now, we use the Alpha argument to validate $[N_R, F_k(N_R, b)]$. Since, the key k is same for all tags, A can not be from Destructive class. Although the binding sequence is randomized, A can not be from Forward class as A does have access to random choice made by R. We start with A in Weak class. Since, the protocol is randomized for R, so the correctness of the protocol (behavior when there is no attacker) depends on the fact that each protocol instance is different. So, in the following we analyze for the two cases: $m'_i = m_i$ and $m'_i \neq m_i$.

1. $[\text{ACCEPT}(N'_R, F_k(N_R, b)]))$:
   The case, $N'_R = N_R$, can only occur $q.2^{-|N_R|}$. The case, $N'_R \neq N_R$, can results in $F_k(N'_R, b) = F_k(N_R, b)$ with probability $q.2^{-S}$

2. $Pr(\text{ACCEPT}([N_R, F_k(N_R, b)']))$:
   The case, $F_k(N_R, b)' = F_k(N_R, b)$, can only occur $q.2^{-|N_R|}$. The case, $F_k(N_R, b)' \neq F_k(N_R, b)$ has $Pr(\text{ACCEPT}) = 0$.

3. $Pr(\text{ACCEPT}([N'_R, F_k(N'_R, b)]))$:
   The case, $[N'_R, F_k(N'_R, b)] = [N_R, F_k(N_R, b)]$, can only occur $q.2^{-|N_R|}$.
   The case, $[N'_R, F_k(N'_R, b)] \neq [N_R, F_k(N_R, b)]$ has $Pr(\text{ACCEPT}) = 0$.

A.3 Π_3: (1) $R \rightarrow \mathbf{ID}_i : a$ (2) $\mathbf{ID}_i \rightarrow R : F_k(a, N_{\mathbf{ID}})$

The binding sequence β_R in this case is $[a, F_k(a, N_{\text{ID}})]$. The corresponding the Beta arguments are as follow.

$$\Theta, \Pi_3, [a, F_k(a, N_{\text{ID}})] \models \text{EXST}(R \rightarrow \text{ID}, \beta_R(i)) \tag{A.7}$$

$$\Theta, \Pi_3, [a, F_k(a, N_{\text{ID}})] \models \text{OPER}(R \rightarrow \text{ID}, \beta_R(i)) \tag{A.8}$$

$$\Theta, \Pi_3, [a, F_k(a, N_{\text{ID}})] \models \text{UNTC}(\text{ID}, \Pi_3) \tag{A.9}$$

R should be able to distinguish between $\beta_R(i) = [a, F_k(a, N_{\text{ID}}(i))]$ and $\beta_R(j) = [a, F_k(a, N_{\text{ID}}(j))]$. This is not possible in as $N_{\text{ID}}(i)$ and $N_{\text{ID}}(j)$ are randomly generated by the tags. So, we include the assumption (a) and (b).

For $\text{OPER}(R \rightarrow \text{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [a, F_k(a, N_{\text{ID}}(i))]$ and $\beta_R(j) = [a, F_k(a, N_{\text{ID}}(j))]$. This is only possible if R remembers all received values for N_{ID}, otherwise R can not distinguish between two instances of the protocol. So, we include the assumption (d). Even so, R does not know the random choices N_{ID}; so, we include the assumption (e).

For $\text{UNTC}(\Pi_3 \rightarrow \text{ID}_i)$, the three instances are as follow: $\beta_R(i) = [a, F_k(a, N_{\text{ID}}(i))]$ (between R and ID_i), $\beta_R(j) = [a, F_k(a, N_{\text{ID}}(j))]$ (between R and ID_i) and

$\beta_R(k) = [a, F_k(a, N_{\text{ID}}(k))]$ (between R and tag_j). A does not know N_{ID}; so, distinguishing is not possible. Thus, Π_2 is private.

We use the Alpha argument to validate the hypothesis $[a, F_k(a, N_{\text{ID}})]$. Since all keys are the same, A can not be from Destructive class. As Π_3 is randomized, A can be from Forward class — A does not know N_R; At the end of attack, A with key k still can not compute $F_k(a, N_{\text{ID}})$. But, with the assumption (e), A can not be Forward because A can invert a response. So, we start with A in Weak class. Note that the protocol is not randomized for R, so unlike Π_2 we only consider one case — $m'_i \neq m_i$

1. $Pr(\text{ACCEPT}([a', F_k(a, N_{\text{ID}})]))$:
   Similar to the corresponding case of Π_1, $a' \neq a$ is not possible.

2. $Pr(\text{ACCEPT}([a, F_k(a, N_{\text{ID}})']))$:
   With $F_k(a, N_{\text{ID}})' = F_k(a, N'_{\text{ID}})$ the event ACCEPT occurs. So, A should be in Coward class.

3. $Pr(\text{ACCEPT}([a', F_k(a', N_{\text{ID}})]))$:
   Similar to the first case, $a' \neq a$ is not possible.

Hence, A is from Coward class.

A.4 Π_4: (1) $R \rightarrow \mathbf{ID}_i : N_R$ (2) $ID_i \rightarrow R : F_k(N_R, N_{\mathbf{ID}})$

The binding sequence in this case is $[N_R, F_k(N_R, N_{\text{ID}})]$. The corresponding Beta argument is as follows.

$$\Theta, \Pi_4, [N_R, F_k(N_R, N_{\text{ID}})] \models \text{EXST}(R \rightarrow \text{ID}, \beta_R(i)) \tag{A.10}$$

$$\Theta, \Pi_4, [N_R, F_k(N_R, N_{\text{ID}})] \models \text{OPER}(R \rightarrow \text{ID}, \beta_R(i)) \tag{A.11}$$

$$\Theta, \Pi_4, [N_R, F_k(N_R, N_{\text{ID}})] \models \text{UNTC}(\text{ID}, \Pi_4) \tag{A.12}$$

As per operational definition of $EXST(R \rightarrow \text{ID}_i)$, R should be able to distinguish between $\beta_R(i) = [N_R(i), F_k(N_R(i), N_{\text{ID}}(i))]$ and $\beta_R(j) = [N_R(j), F_k(N_R(j), N_{\text{ID}}(j))]$. This is not possible in honest model as all the keys are same. Similar to Π_1, Π_2 and Π_3, we include the assumptions (a) and (b). Since, R does not know the random choices N_{ID}, so we further assume (e) $F_k(...)$ is a PRP with an efficiently computable inverse function $F_k^{-1}(...)$.

For $\text{OPER}(R \rightarrow \text{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [N_R(i), F_k(N_R(i), N_{\text{ID}}(i))]$ and $\beta_R(j) = [N_R(j), F_k(N_R(j), N_{\text{ID}}(j))]$. With the assumption (e) this is possible due to random challenge N_R.

For $\text{UNTC}(\Pi_4 \rightarrow \text{ID}_i)$, adversary A should be able to pick a pair with a probability different from the rest of two pair. The three instances of protocol transcripts are as follow: $\beta_R(i) = [N_R(i), F_k(N_R(i), N_{\text{ID}}(i))]$ (between R and ID_i), $\beta_R(j) = [N_R(j), F_k(N_R(j), N_{\text{ID}}(j))]$ (between R and ID_i) and $\beta_R(k) =$

$[N_R(k), F_k(N_R(k), N_{\mathrm{ID}}(k))]$ (between R and ID_j). As A does not know N_{ID} and N_R, the distinguishing is not possible Thus, Π_4 is private.

We use the Alpha argument to validate the hypothesis $[N_R, F_k(N_R, N_{\mathrm{ID}})]$. Since, we assume that tag stores the key that is same for all tags, so the binding sequence can not be validated for any adversary from Destructive class. Π_4 is randomized so A can be from Forward class, because A does not have access to random choice made by ID_i, namely N_R. But, with the assumption (e), A can compute N_R. So, we start with A in Weak class.

1. $Pr(\mathrm{ACCEPT}([N'_R, F_k(N_R, N_{\mathrm{ID}})]))$:
   We consider the two cases: $N'_R = N_R$ from different instance; and $N'_R \neq N_R$. The probability for the former case is $q.2^{-|N_R|}$. The probability for the later case (and still obtaining $F_k(N_R, N_{\mathrm{ID}})$) is approximately $q.2^{-S}$.

2. $Pr(\mathrm{ACCEPT}([N_R, F_k(N_R, N_{\mathrm{ID}})']))$:
   We consider the two cases: $F_k(N_R, N_{\mathrm{ID}})' = F_k(N_R, N_{\mathrm{ID}})$ from a different instance; and $F_k(N_R, N_{\mathrm{ID}})' \neq F_k(N_R, N'_{\mathrm{ID}})$. The probability of first case is $q.2^{-S}$. The second case can occur if A simply relays N_R to another tag. So we need to assume A is from Coward class.

3. $Pr(\mathrm{ACCEPT}([N'_R, F_k(N_R, N_{\mathrm{ID}})']))$:
   We consider the two cases: $[N'_R, F_k(N_R, N_{\mathrm{ID}})'] = [N_R, F_k(N_R, N_{\mathrm{ID}})]$, i.e., replay of an instance that occurs with a different tag; and $[N'_R, F_k(N_R, N_{\mathrm{ID}})'] \neq [N_R, F_k(N_R, N_{\mathrm{ID}})]$. The probability of first case is $q.2^{-S}$. The second case can occur if A simply relays any new N'_R to another tag. So we need to assume A is from Coward class.

Hence, A is from Coward class.

A.5 Π_5: (1) $R \rightarrow \mathbf{ID}_i : a$ (2) $\mathbf{ID}_i \rightarrow R : F_{K_i}(a, b)$

First we consider the Beta argument. β_R in this case is $[a, F_{K_i}(a, b)]$

$$\Theta, \Pi_2, [a, F_{K_i}(a, b)] \models \mathrm{EXST}(R \rightarrow \mathrm{ID}, \beta_R(i)), \tag{A.13}$$

$$\Theta, \Pi_2, [a, F_{K_i}(a, b)] \models \mathrm{OPER}(R \rightarrow \mathrm{ID}, \beta_R(i)), \tag{A.14}$$

$$\Theta, \Pi_2, [a, F_{K_i}(a, b)] \models \mathrm{UNTC}(\mathrm{ID}, \Pi_5) \tag{A.15}$$

For $\mathrm{EXST}(R \rightarrow \mathrm{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [a, F_{K_i}(a, b)]$ and $\beta_R(j) = [a, F_{K_j}(a, b)]$. Such a distinguishing is possible as K_i is unique for each tag.

For $\mathrm{OPER}(R \rightarrow \mathrm{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [a, F_{K_i}(a, b)]$ and $\beta_R(j) = [a, F_{K_i}(a, b)]$. Such a distinguishing is not possible; so, we include the assumption (c).

For UNTC(ID, Π_2), adversary A should be able to pick a pair with a probability different from the rest of two pair. As per operational definition of untraceability, the three instances of binding sequence are as follow: $\beta_R(i) = [a, F_{K_i}(a,b)]$ (between R and ID$_i$), $\beta_R(j) = [a, F_{K_i}(a,b)]$ (between R and ID$_i$) and $\beta_R(k) = [a, F_{K_j}(a,b)]$ (between R and ID$_j$). Clearly, this is possible as first two instances are exactly same and different from the third one. So, we need to assume that (h) privacy is of no concern for the items under consideration.

Now, we use the Alpha argument to validate $[a, F_{K_i}(a,b)]$. Since, we assume that tag stores the key is different for all tags, we start validating the binding sequence from A in Destructive class.

1. $Pr(\text{ACCEPT}([a', F_{K_i}(a,b)]))$:
   The case $a' \neq a$ is not possible.

2. $Pr(\text{ACCEPT}([a, F_{K_i}(a,b)']))$:
   The event ACCEPT occurs if $F_{K_i}(a,b)' = F_{K_j}(a,b)$, for $K_i \neq K_j$. This could happen with probability $n.2^{-|K|}$ (n is total number of tags), which is negligible.

3. $Pr(\text{ACCEPT}([a', F_{K_i}(a,b)']))$:
   The case that involves $a' \neq a$ is not possible.

Hence, A is Destructive class.

A.6 Π_6: (1) $R \rightarrow \mathbf{ID} : N_R$ (2) $\mathbf{ID} \rightarrow R : F_K(N_R, b)$

In this case, $\beta_R = [N_R, F_{K_i}(N_R, b)]$.

$$\Theta, \Pi_6, [N_R, F_{K_i}(N_R,b)] \models \text{EXST}(R \rightarrow \text{ID}, \beta_R(i)), \tag{A.16}$$

$$\Theta, \Pi_6, [N_R, F_{K_i}(N_R,b)] \models \text{OPER}(R \rightarrow \text{ID}, \beta_R(i)), \tag{A.17}$$

$$\Theta, \Pi_6, [N_R, F_{K_i}(N_R,b)] \models \text{UNTC}(\text{ID}, \Pi_6) \tag{A.18}$$

For EXST($R \rightarrow$ ID, $\beta_R(i)$), R should be able to distinguish between $\beta_R(i) = [N_R(i), F_{K_i}(N_R(i), b)]$ and $\beta_R(j) = [N_R(j), F_{K_j}(N_R(j), b)]$. Clearly, such a distinguishing is possible as K_i is unique for each tag.

For OPER($R \rightarrow$ ID, $\beta_R(i)$), R should be able to distinguish between $\beta_R(i) = [N_R(i), F_{K_i}(N_R(i), b)]$ and $\beta_R(j) = [N_R(j), F_{K_i}(N_R(j), b)]$. Clearly, such a distinguishing is possible as N_R is unique in each query, except with a negligible probability of $2^{-|N_R|}$.

As per operational definition of untraceability, the three instances of binding sequence are as follow: $\beta_R(i) = [N_R(i),\ F_{K_i}(N_R(i), b)]$ (between R and ID$_i$), $\beta_R(j) = [N_R(j),\ F_{K_i}(N_R(j), b)]$ (between R and ID$_i$) and $\beta_R(k) =$

$[N_R(k),\ F_{K_j}(N_R(k), b)]$ (between R and ID_j). Clearly, this is not possible for adversary as all three responses are randomized due to N_R.

Now, we use the Alpha argument to validate $[N_R, F_K(N_R, b)]$. Since, we assume that tag stores the key is different for all tags, so we start with A in Destructive class.

1. $Pr(\mathrm{ACCEPT}([N'_R, F_K(N_R, b)]))$:
   We now consider two cases. The first case is $N'_R = N_R$ but N'_R is from different instance. The second case is $N'_R \neq N_R$. The first case can occur with a probability $q.2^{-|N_R|}$. The second case can produce $F_K(N_R, b)]$ with a probability $q.2^{-S}$.

2. $Pr(\mathrm{ACCEPT}([N_R, F_K(N_R, b)']))$:
   The possible construction of $F_K(N_R, b)' \neq F_K(N_R, b)$ that can produce ACCEPT is $F_K(N_R, b)' = F_{K'}(N_R, b)$, which can occur with a negligible probability of $n.2^{-|K|}$. The probability, $Pr(F_K(N_R, b)' = F_K(N_R, b))$ from a different instance is approximately $n.2^{-S} + q.2^{-|N_R|}$.

3. $Pr(\mathrm{ACCEPT}([N'_R, F_K(N_R, b)'] \wedge ACCEPT)$:
   There are two cases here. If $[N'_R, F_K(N_R, b)'] = [N_R, F_K(N_R, b)]$ then this is the case of replay from different binding sequence. Probability that such a case occurs is approximately $n.2^{-S} + q.2^{-|N_R|}$. The other case, $[a', F_K(a, N_{\mathrm{ID}})'] \neq [a, F_K(a, N_{\mathrm{ID}})]$ can result in ACCEPT event with a probability of $n.2^{-|K|}$.

Hence, A is in Destructive class.

A.7 Π_7: (1) $R \rightarrow \mathbf{ID}_i : a$ (2) $\mathbf{ID}_i \rightarrow R : F_K(a, N_{\mathbf{ID}})$

For this construction, $\beta_R = [a, F_{K_i}(a, N_{\mathrm{ID}})]$ The Beta arguments are as follow.

$$\Theta, \Pi_7, [a, F_{K_i}(a, N_{\mathrm{ID}})] \models \mathrm{EXST}(R \rightarrow \mathrm{ID}, \beta_R(i)), \tag{A.19}$$

$$\Theta, \Pi_7, [a, F_{K_i}(a, N_{\mathrm{ID}})] \models \mathrm{OPER}(R \rightarrow \mathrm{ID}, \beta_R(i)), \tag{A.20}$$

$$\Theta, \Pi_7, [a, F_{K_i}(a, N_{\mathrm{ID}})] \models \mathrm{UNTC}(\mathrm{ID}, \Pi_7) \tag{A.21}$$

For $\mathrm{EXST}(R \rightarrow \mathrm{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [a, F_{K_i}(a, N_{\mathrm{ID}}(i))]$ and $\beta_R(j) = [a, F_{K_j}(a, N_{\mathrm{ID}}(j))]$. Theoretically, such a distinguishing is possible as K_i is unique for each tag. Practically, there is another problem; R does not know the random choice N_{ID} made by the tag. Thus, (e) $F_K(...)$ should be an PRP with an efficiently computable $F_K^{-1}(...)$.

For $\mathrm{OPER}(R \rightarrow \mathrm{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [a, F_{K_i}(a, N_{\mathrm{ID}}(i))]$ and $\beta_R(j) = [a, F_{K_i}(a, N_{\mathrm{ID}}(j))]$. Even with the assumption (e), such a distinguishing is not possible as R ca not predict the random choice N_{ID} made by the tag. So, we need to assume (d) reader query

a tag more than once and detects the collisions in the random values obtained from the tag.

The three instances of binding sequence are as follow: $\beta_R(i) = [a, F_{K_i}(a, N_{\mathrm{ID}}(i))]$ (between R and ID_i), $\beta_R(j) = [a,\ F_{K_i}(a, N_{\mathrm{ID}}(j))]$ (between R and ID_i) and $\beta_R(k) = [a,\ F_{K_j}(a, N_{\mathrm{ID}}(k))]$ (between R and ID_j). Clearly, this is not possible.

Now, we use the Alpha argument to validate $[a, F_{K_i}(a, N_{\mathrm{ID}})]$. Since each tag has its own unique key, so we start with A in Destructive class.

1. $Pr(\mathrm{ACCEPT}([a', F_{K_i}(a, N_{\mathrm{ID}})]))$:
   The case $a' \neq a$ can not occur.

2. $Pr(\mathrm{ACCEPT}([a, F_{K_i}(a, N_{\mathrm{ID}})']))$:
   There are two cases here:
   $[a, F_{K_i}(a, N_{\mathrm{ID}})'] = [a, F_{K_i'}(a, N_{\mathrm{ID}})]$ and $[a, F_{K_i}(a, N_{\mathrm{ID}})'] = [a, F_{K_i}(a, N'_{\mathrm{ID}})]$.
   Both of these cases occur with a negligible probability.

3. $Pr(\mathrm{ACCEPT}([a', F_{K_i}(a, N_{\mathrm{ID}})'] \wedge ACCEPT)$:
   This case is similar to the previous case.

Hence, A can be from Destructive class.

A.8 Π_8: (1) $R \rightarrow \mathbf{ID}_i : N_R$ (2) $\mathbf{ID}_i \rightarrow R : F_K(N_R, N_{\mathbf{ID}})$

We start with the Beta argument. β_R in this case is $[N_R, F_{K_i}(N_R, N_{\mathrm{ID}})]$ We need to show the validity of following arguments.

$$\Theta, \Pi_8, [N_R, F_{K_i}(N_R, N_{\mathrm{ID}})] \models \mathrm{EXST}(R \rightarrow \mathrm{ID}, \beta_R(i)), \tag{A.22}$$

$$\Theta, \Pi_8, [N_R, F_{K_i}(N_R, N_{\mathrm{ID}})] \models \mathrm{OPER}(R \rightarrow \mathrm{ID}, \beta_R(i)), \tag{A.23}$$

$$\Theta, \Pi_8, [N_R, F_{K_i}(N_R, N_{\mathrm{ID}})] \models \mathrm{UNTC}(\mathrm{ID}, \Pi_8) \tag{A.24}$$

For $\mathrm{EXST}(R \rightarrow \mathrm{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [N_R(i), F_{K_i}(N_R(i), N_{\mathrm{ID}}(i))]$ and $\beta_R(j) = [N_R(j), F_{K_j}(N_R(j), N_{\mathrm{ID}}(j))]$. Such a distinguishing is possible as K_i is unique for each tag. Since R does not know the random choice N_{ID} made by the tag, we include the assumption (e).

For $\mathrm{OPER}(R \rightarrow \mathrm{ID}, \beta_R(i))$, R should be able to distinguish between $\beta_R(i) = [N_R(i), F_{K_i}(N_R(i), N_{\mathrm{ID}}(i))]$ and $\beta_R(j) = [N_R(j), F_{K_i}(N_R(j), N_{\mathrm{ID}}(j))]$. Clearly, such a distinguishing is possible as N_R is different for each tag, but we also need to include the assumption (e).

For $\mathrm{UNTC}(\mathrm{ID}, \Pi_8)$, adversary A should be able to pick a pair with a probability different from the rest of two pair. As per operational definition of untraceability, the three instances of binding sequence are as follow: $\beta_R(i) = [N_R(i),\ F_{K_i}(N_R(i), N_{\mathrm{ID}}(i))]$ (between R and $\mathrm{ID}D_i$), $\beta_R(j)$

$= [N_R(j),\ F_{K_i}(N_R(j), N_{\mathrm{ID}}(j))]$ (between R and ID_i) and $\beta_R(k) = [N_R(k),\ F_{K_k}(N_R(k), N_{\mathrm{ID}}(k))]$ (between R and ID_j). Clearly, this is not possible for adversary.

We start by assuming A in Destructive class.

1. $Pr(\mathrm{ACCEPT}([N'_R, F_K(N_R, N_{\mathrm{ID}})]))$:
   The first case is $N'_R = N_R$ with N'_R from a different instance. The second case is $N'_R \neq N_R$ but generates $F_K(N_R, N_{\mathrm{ID}})$. The first case can occur with a probability $q.2^{-|N_R|}$. The second case can occur with a probability $n.2^{-S}$

2. $Pr(\mathrm{ACCEPT}([N_R, F_K(N_R, N_{\mathrm{ID}})']))$:
   The first case, $Pr(F_K(N_R, N_{\mathrm{ID}})' = F_K(N_R, N_{\mathrm{ID}}))$ from a different sequence, can occur with $n.2^{-S}$. The second case, $F_K(N_R, N_{\mathrm{ID}})' \neq F_K(N_R, N_{\mathrm{ID}})$ can lead to ACCEPT for $F_{K'}(N_R, N_{\mathrm{ID}})$ and $F_K(N_R, N'_{\mathrm{ID}})$; both of them have a negligible probability of occurrence.

3. $Pr(\mathrm{ACCEPT}([N'_R, F_K(N_R, N_{\mathrm{ID}})'] \wedge ACCEPT)$:
   The analysis is similar to the previous case.

Hence A can be from Destructive class.